AMENDMENTS TO THE CLAIMS:

The text of all pending claims, (including withdrawn claims) is set forth below. Canceled and not entered claims are indicated with claim number and status only. The claims as listed below show added text with underlining and deleted text with ~~strikethrough~~. The status of each claim is indicated with one of (original), (currently amended), (canceled), (withdrawn), (new), (previously presented), or (not entered).
Applicants reserve the right to pursue any canceled claims at a later date.
The following listing of claims will replace all prior versions, and listings, of claims in the application:

1-6. (canceled)

7. (currently amended) A method for managing data in a source file ~~of the type from which data may be extracted for use in an application~~ wherein the source file data is described by an extensible markup language, the method comprising:

providing a source file described in an extensible markup language;

structuring the data in the source file in the form of objects, wherein components of the objects ~~can be~~are stored in first files, wherein the components each represent a logical unit of an object; ~~and~~

providing a second file having a first mechanism for referencing the components as a higher-order, object-based logical level for storing; and selectively directly accessing the objects, the foregoing providing hierarchical structuring of object complexes and distribution of data of objects among a plurality of files to enable a reading-in tool to pass over or avoid having to read or process portions of the source file data when seeking other portions of the source file data for use in the application~~.~~; and

providing the reading-in tool, wherein the reading-in tool passes over or avoids reading or processing portions of the source file data when seeking other portions of the source file data for use in the application.

8. (previously presented) The method according to claim 7, wherein the components are themselves objects.

9. (previously presented) The method according to claim 7, wherein the components are stored in object-specific generic containers, and wherein the containers are provided for referencing the respective object.

10. (previously presented) The method according to claim 7, wherein the extensible markup language is XML.

11. (currently amended) A system for managing data in a source file of the type from which data may be extracted for use in an application wherein the source file data is described by an extensible markup language, wherein objects for structuring the data are provided, wherein components of the objects can be stored in first files, wherein the components each represent a logical unit of an object, and wherein a second file having first mechanisms for referencing the components is provided as a higher-order, object-based logical level for storing the objects, wherein a parser is provided to read the source file, wherein the first files and the second file are arranged in a hierarchical structure with a distribution of data of objects among a plurality of the first files to enable the ~~a reading in tool~~parser to pass over or avoid having to read or process portions of the source file data when seeking other portions of the source file data for use in the application.

12. (previously presented) The system according to claim 11, wherein the components are themselves objects.

13. (previously presented) The system according to claim 11, wherein object-specific generic containers are provided for storing the components of the objects, with said containers serving to reference the respective object.

14. (previously presented) The system according to claim 11, wherein the extensible markup language is XML.

15. (new) The method according to claim 7, wherein the reading-in tool is a parser.

16. (new) The method according to claim 15, wherein the parser is a XML-parser.

17. (new) The method according to claim 7, wherein the reading-in tool passes over data beginning from a certain start tag up to the associated end tag.

18. (new) The method according to claim 7, wherein parts of objects are distributed among a plurality of files, wherein a core information necessary for identifying the object and its type is present in a source file and wherein the object's actual useful information is relocated to a relocation file.

19. (new) The method according to claim 7, wherein references to relocated objects contain an object identification data, data regarding a target file in which the object is located and object identification data in the target file.

20. (new) The method according to claim 19, wherein the object identification data in the target file is an object ID and an object name.

21. (new) The system according to claim 11, wherein parts of objects are distributed among a plurality of files, wherein a core information necessary for identifying the object and its type is present in a source file and wherein the object's actual useful information is relocated to a relocation file.

22. (new) The system according to claim 11, wherein references to relocated objects contain an object identification data, data regarding a target file in which the object is located and object identification data in the target file.

23. (new) The system according to claim 11, wherein the object identification data in the target file is an object ID and an object name.